

# Managing PowerShell Gallery

...the pragmatic way

# Agenda

- About PowerShell (Gallery) repository
- Why having an own repository?
- Types of repositories
- Making an -internal- repo effective
- Q&A / Discussions

# Get-Person -Short

Andreas Bellstedt



@AndiBellstedt



@andibellstedt.bsky.social



github.com/AndiBellstedt



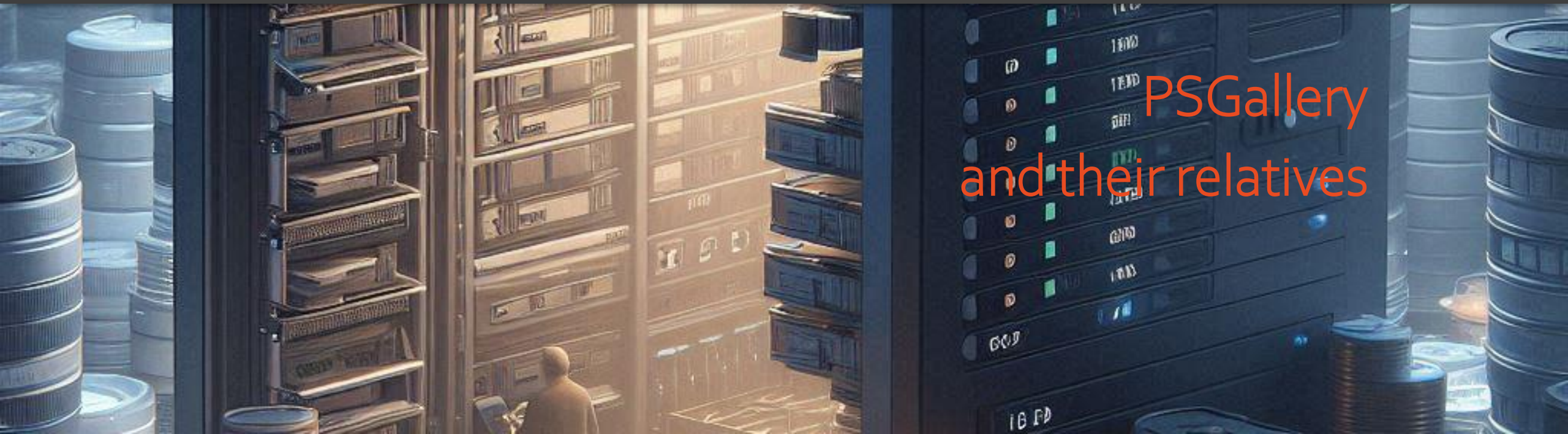
<https://www.linkedin.com/in/andreas-bellstedt-1862768>

IT infrastructure guy and PowerShell fellow

Microsoft Infrastructure  
Security



# PowerShell repositories



PSGallery  
and their relatives

# Why care about repositories

- Part of the PS ecosystem
- Easy and convenient, well established
- Pros and Cons in utilization of builtin PSGallery





# Why having an internal repository

- Air gapped systems
- Own modules / internal code
- Control
- Compliance







# Type of repositories





# Repo flavors

## Share based

- Easy to do (relatively)
- No additional software required
- PSGetV2 compatible
- Limited scale
- Access control based on SMB/NTFS permissions
- Potential authentication challenges



## Web based

- More flexible
- Statistics and content infos
- Larger scale
- Access control is web-based (API-Keys, SAML, ...)
- Requires additional Software onprem or SaaS
- Potential requirement of PSGetV3





# Repo flavors - examples

## Share based

- Windows File Services
- NAS systems

## Web based

- NuGet Repositories
  - PSGallery
  - NuGet.org
  - Artifactory
  - Nexus Repository Manager
  - ProGet
- Azure Artifacts\*
  - Azure DevOps
  - Azure Container Registry

\*) requires PowerShellGetv3

A man with a beard and dark hair, wearing a red and black checkered shirt, is smiling and holding a large, light-colored wooden plank diagonally across his chest. He is in a workshop or garage with various tools and equipment visible in the background. The lighting is warm and the background is slightly blurred.

Making an -internal- repo effective

# Various perspectives - why effective?

- Efficiency/convenience vs. Effectiveness
- Necessarily
  - Having it, does not mean knowing it
  - Knowing it, does not mean using it



# Various perspectives - why effective?

- For your “Customers”
  - Embrace the SystemProfile
- Stay current / solve the airgap problem
  - Approved module list
  - Proxy system

# Demo

(also called... the switch into the interesting part)

# Conclusion

- Caring about PSGallery may be worth a consideration
- Distribution of own modules and Updates across the environment easier
  - Consider breaking changes in module Updates!
- Building a Repo on a server is only half of the story
  - Bring it to your customers!
- You can start small without additional cost and software
- Be advised, PS5 and PS7 are different products
  - They also have different repositories and module directories



# Question & Answers



# Ressources

- <https://learn.microsoft.com/en-us/powershell/gallery/how-to/working-with-local-psrepositories>
- <https://inedo.com/proget>
- <https://devblogs.microsoft.com/powershell/psresourceget-is-generally-available>
- <https://devblogs.microsoft.com/scripting/understanding-the-six-powershell-profiles>

– Thank you –

Andreas Bellstedt

 @AndiBellstedt

 [github.com/AndiBellstedt](https://github.com/AndiBellstedt)